

VEHICLE ROUTING PROBLEM UNTUK DISTRIBUSI BARANG MENGUNAKAN ALGORITMA SEMUT

Agung Hadhiatma^{1*}, Alexander Purbo^{2*}

^{1,2} Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma
Paingan, Maguwoharjo, Depok, Sleman 55282

Email: ^{1*} agunghad@usd.ac.id, ^{2*} purboalexander@gmail.com

Abstrak

Vehicle routing problem (VRP) adalah masalah optimasi kombinatorial penentuan rute yang dapat diterapkan pada pengaturan pendistribusian barang. Penelitian ini menguji salah satu algoritma metaheuristik yaitu algoritma semut pada VRP untuk menentukan rute yang minimal dalam pendistribusian barang. Dalam penelitian ini, tipe vehicle routing problem yang klasik cukup didekati dengan TSP (travelling sales problem). Dibandingkan dengan algoritma konvensional, algoritma metaheuristik membutuhkan waktu komputasi yang lebih cepat, namun hanya menghasilkan pendekatan dari sebuah solusi optimal. Penelitian ini bermaksud selain untuk membuat sebuah solusi aplikasi untuk kasus pendistribusian barang dari gudang ke beberapa toko pada PT CircleK regional Yogya, juga untuk menguji sejauh mana tingkat akurasi hasil rute terpendek dari algoritma semut untuk penyelesaian kasus tersebut. Hasil penelitian menunjukkan, untuk nilai parameter tertentu, semakin banyak input jumlah toko, tingkat akurasi rute terpendek dari algoritma semut akan berkurang. Namun dengan penempatan parameter jumlah semut yang tepat, tingkat akurasi algoritma semut akan meningkat. Menentukan parameter jumlah semut yang tepat menjadi sebuah tantangan, karena bila kurang akan mengurangi tingkat akurasi sedangkan bila lebih akan menambah beban komputasi.

Kata kunci: *vehicle routing problem, algoritma semut, tingkat akurasi, parameter jumlah semut*

1. PENDAHULUAN

Dalam proses distribusi barang, sebuah perusahaan mengirimkan barang dari suatu lokasi (pabrik, gudang) ke beberapa tempat tujuan (distributor, toko) dengan menggunakan kendaraan (mobil box). Salah satu tujuan dari proses distribusi barang adalah meminimalkan biaya pengiriman. Hal ini dikenal sebagai *vehicle routing problem* (VRP). Tujuan dari VRP adalah mendistribusikan barang dengan biaya minimum melalui rute tertentu dari suatu sumber menuju ke beberapa simpul (*node*) (Cordeau dkk, 2007). Rute tersebut berawal dari sumber dan berakhir di sumber yang sama.

VRP adalah sebuah masalah optimasi kombinatorial atau masalah pemrograman integer yang masuk kategori *NP-Hard Problem*. Dalam *NP-Hard Problem*, usaha komputasi yang digunakan akan semakin meningkat seiring dengan meningkatnya ruang lingkup masalah (Christos dan Steiglitz, 1982). Kasus NP-Hard dapat diselesaikan dengan menggunakan metode konvensional (brute force) dan metaheuristik (Pop dkk, 2011). Metode brute force akan membangkitkan semua kemungkinan solusi yang ada, kemudian dari semua solusi tersebut dipilih yang terbaik. Sedangkan metode metaheuristik memberikan perkiraan solusi yang mendekati optimal, sehingga proses perhitungan menjadi lebih cepat daripada metode konvensional (Glover dan Kochenberger, 2003). Salah satu dari algoritma metaheuristik adalah sebuah penyelesaian menggunakan *ant colony optimization*.

Ant colony optimization yang diselesaikan dengan algoritma semut diinspirasi oleh sebuah koloni semut ketika mencari sumber makanan. Semut-semut tersebut menemukan jarak terpendek antara sarang semut dan sumber makanannya berdasarkan intensitas jejak semut (*pheromone*). Algoritma ini pertama kali diperkenalkan oleh Dorigo dkk (1996). Beberapa penelitian telah dilakukan untuk memperbaiki efektivitas algoritma semut ini, seperti yang telah dilakukan oleh Hung (2007) dan Wei (2010).

Ada berbagai tipe masalah pada VRP. Permasalahan VRP klasik dapat diselesaikan dengan pendekatan TSP (*travelling sales problem*). TSP adalah penentuan rute terpendek dengan urutan

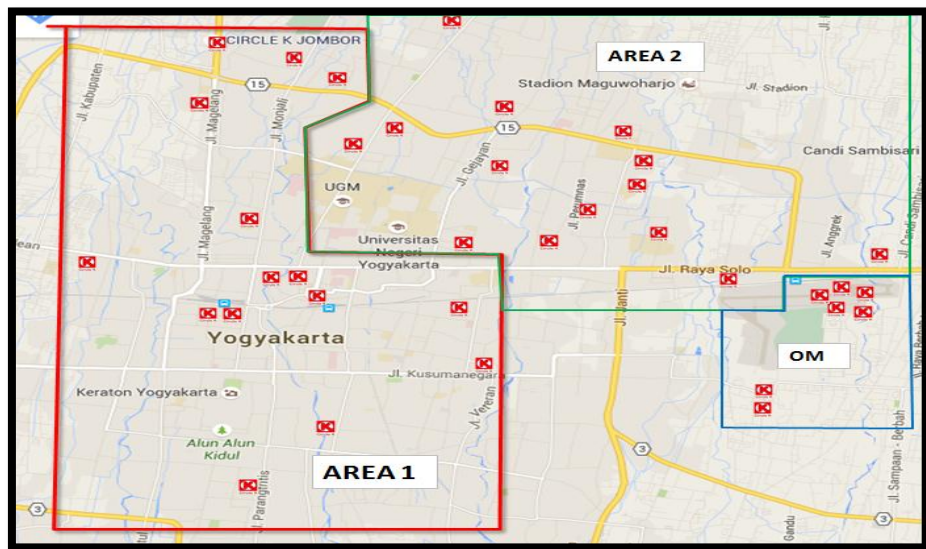
tertentu pada *node-node*, dengan sebuah batasan (*constrain*). Salah satu contoh dari batasan tersebut adalah: setiap *node* hanya boleh dikunjungi satu kali (Lawler,1985).

Dalam penelitian ini, traveling sales problem digunakan untuk meminimalkan rute pendistribusian barang pada PT CircleK Indonesia Utama regional Yogyakarta. Algoritma metaheuristik yang dipilih untuk menyelesaikan masalah ini adalah algoritma semut. Permasalahan distribusi pada PT CircleK adalah bagaimana mengatur rute pengiriman barang dari gudang ke sejumlah toko PT CircleK yang tersebar di berbagai tempat di Yogyakarta secara efisien. Kantor regional Yogyakarta PT CircleK meliputi 1 gudang dan 34 toko. Sebuah kendaraan dari perusahaan dalam satu hari mengirimkan barang dari gudang ke paling sedikit 5 toko. Informasi yang dihasilkan adalah rute jalan yang harus ditempuh dari gudang ke beberapa toko yang menjadi tujuan pengiriman.

Penelitian ini selain untuk menyediakan informasi rute dengan jarak yang minimum untuk distribusi barang di kantor regional PT CircleK Yogya, juga untuk menganalisa efektivitas/akurasi algoritma semut terhadap penyelesaian kasus distribusi barang tersebut. Algoritma semut yang digunakan dan dianalisa mengacu pada usulan dari Dorigo dkk (1996). Pemasalahan dan penyelesaian distribusi barang (VRP) dimodelkan dengan TSP.

2. METODOLOGI

Hal pertama yang dilakukan dalam penelitian adalah mengambil data lokasi gudang dan toko PT.CircleK Indonesia Utama region Yogyakarta. Pendistribusian barang dimulai dari Gudang milik PT CircleK Indonesia Utama regional Yogyakarta di Jl. Raya Magelang KM 7 No. 9 Sendangadi Mlati Sleman. Berikut adalah peta dari lokasi toko (gambar1).



Gambar 1 Peta Persebaran Toko pada PTCircleK Region Yogyakarta

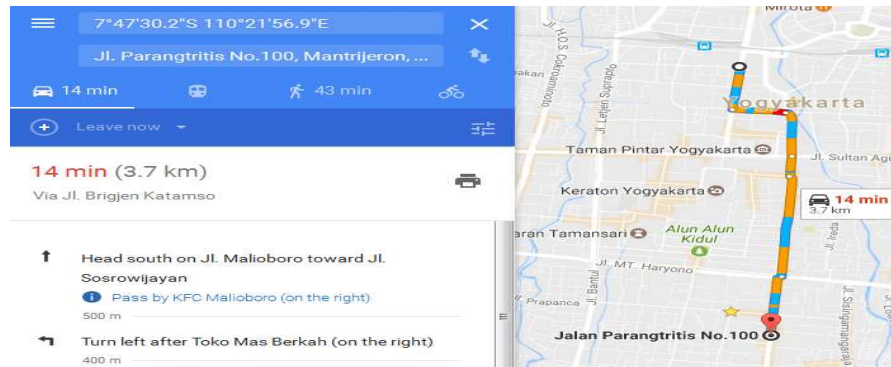
Data peta lokasi toko merupakan sebuah himpunan relasi antara 2 toko dengan atribut kode toko (berelasi ke data alamat dan koordinat lokasi toko), jarak terpendek antara 2 toko, dan rute jalan terpendek antara 2 toko.

Tabel 1: Data peta lokasi toko.

Kode Toko Awal	Kode Toko Tujuan	Jarak terpendek 2 toko	Rute terpendek 2 toko
YOG0101	YOG0102
dst	Dst	Dst	Dst

Data lokasi, jarak dan jalur, didapatkan dengan bantuan aplikasi Google Maps. Input data koordinat toko dihasilkan berdasarkan alamatnya masing-masing. Koordinat gudang dan masing-

masing toko digunakan untuk mendapatkan rute dan jarak dari dua titik kooordinat yang berbeda. Data yang didapat disimpan dalam format pada tabel 1. Berikut ini adalah peta data lokasi toko, rute dan jarak antar toko pada gambar 2 yang merepresentasikan tabel 1.



Gambar 2. Pencarian rute dengan Google Map

Langkah berikutnya adalah menentukan rute distribusi menggunakan prinsip TSP (*travelling sales problem*) yang diselesaikan dengan algoritma semut. Algoritma semut mempunyai aturan dasar sebagai berikut:

- Setiap semut akan mengunjungi kota yang belum pernah dikunjungi sebelumnya. Hal ini diimplementasikan dengan cara menyimpan semua kota yang telah dikunjungi (pada tabu list) sampai semut tersebut menyelesaikan seluruh perjalanannya, dan akan mereset kembali tabulist tersebut setelah semut itu menyelesaikan satu siklus perjalanan.
 - Kunjungan semut dari kota i ke kota j , pada interval antara t dan $(t+1)$, berdasarkan probabilitas yang bergantung pada fungsi jarak $d(C_i, C_j)$ dan jumlah intensitas jejak semut (*pheromone*) pada jalur yang menghubungkan antara C_i dan C_j
 - Setelah selesai dalam satu siklus, semut itu akan meninggalkan jejak *pheromone* pada setiap sisi yang dilaluinya.
 - Proses akan berulang sampai hasil konvergen atau telah mencapai iterasi tertentu.
- Probabilitas transisi dari kota i ke kota j diberikan dengan persamaan 1.

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{i \in dii} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} \quad (1)$$

Parameter α dan β digunakan untuk mengendalikan tingkat kepentingan relatif dari jejak dan visibilitas. τ_{ij} adalah intensitas jejak semut pada *edge*(i, j) pada saat t , maka setelah suatu semut menyelesaikan perjalanan dalam suatu siklus, rumus perubahan intensitas jejak berdasarkan persamaan 2.

$$\tau_{ij}(t+N) = \rho \tau_{ij}(t) + \Delta \tau_{ij}(t, t+N) \quad (2)$$

dengan ρ adalah koefisien dengan nilai 0 sampai 1, dimana:

$$\Delta \tau_{ij}(t, t+N) = \sum_{k=1}^m \Delta \tau_{ij}^k(t, t+N) \quad (3)$$

$\sum_{k=1}^m \Delta \tau_{ij}^k(t, t+N)$ adalah jejak yang ditinggalkan oleh semut k pada *edge* (i, j) pada saat antara t sampai $(t+1)$, yang dihitung berdasarkan persamaan 4, sebagai berikut:

$$\Delta \tau_{ij}^k(t, t+N) = \begin{cases} \frac{Q}{L^k}; & \text{jika semut } k \text{ menggunakan } \textit{edge}(i, j) \\ & \text{dalam perjalanannya} \\ 0; & \text{jika semut } k \text{ tidak menggunakan } \textit{edge}(i, j) \end{cases} \quad (4)$$

Dimana Q = Konstanta, L^k = panjang jalur untuk perjalanan semut k . Tabu list digunakan untuk menyimpan nama kota yang telah dikunjungi oleh semut. Kota yang sudah dilewati semut akan disimpan di tabu list dan tidak boleh dikunjungi lagi.

Urutan algoritma semut adalah sebagai berikut:

1. Inisialisasi Parameter α (tetapan intensitas jejak semut), β (tetapan visibilitas), τ , m (jumlah semut), iterasi, dan Q (konstanta).
2. Letakan semut ke salah satu node secara acak.
3. Selama dalam iterasi lakukan langkah 4- 6
4. Untuk setiap semut, lakukan :
 - Menghitung probabilitas transisi (persamaan 1) dan memilih probabilitas terbesar untuk bergerak ke node berikutnya
 - Tambahkan node yang sudah dilalui kedalam tabu list
 - Ulangi hingga semut menyelesaikan solusi
5. Untuk setiap semut yang telah menyelesaikan solusi lakukan:
Update feromon τ (persamaan 2) untuk tiap node yang dilalui semut.
6. Jika solusi lokal terbaik lebih baik daripada solusi global
Simpan solusi lokal terbaik sebagai solusi global
7. Selesai.

Langkah penelitian untuk menguji efektivitas algoritma semut untuk menyelesaikan permasalahan distribusi barang dengan pendekatan TSP adalah sebagai berikut:

1. Pengujian algoritma algoritma semut untuk input 6 toko. Hasil rute dan jarak ditampilkan dalam peta.
2. Pengujian algoritma semut untuk masukan toko = 5,6,7,8,9,10. Hasil rute dan jarak dibandingkan dengan algoritma brute force.
3. Pengujian algoritma semut untuk masukan toko = 5,6,7,8,9,10 dengan jumlah semut = 5,6,7,8.
Hasil dan rute jarak dibandingkan dengan algoritma brute force.

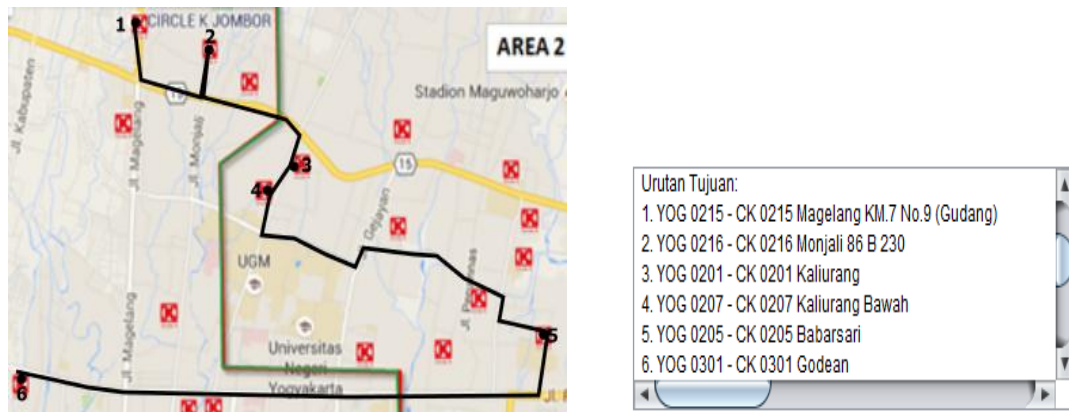
Algoritma brute force digunakan sebagai pembanding karena menghasilkan global optimum. Pembandingan hasil antara algoritma semut dan algoritma brute force bertujuan untuk mengetahui sejauh mana algoritma semut mempunyai hasil yang mendekati dengan algoritma brute force.

3. HASIL DAN PEMBAHASAN

Percobaan pencarian rute pengiriman barang dengan menggunakan algoritma semut terdiri dari 3 langkah. Percobaan pertama adalah untuk melihat sejauh mana algoritma semut dapat diterapkan pada aplikasi penentuan rute distribusi barang dengan input sebanyak n toko. Hasil percobaan adalah jalur rute pengiriman beserta peta. Percobaan berikut adalah menguji akurasi hasil algoritma semut untuk rute distribusi. Pengujian dengan membandingkan hasil algoritma semut dengan algoritma brute force. Percobaan ini dilakukan dengan pendekatan TSP. Percobaan terakhir adalah melihat pengaruh perubahan parameter jumlah semut (m) terhadap efektivitas algoritma semut untuk masalah distribusi barang.

3.1. Percobaan 1.

Percobaan pertama adalah pengujian aplikasi penentuan rute distribusi barang pada PT CircleK menggunakan algoritma semut. Parameter awal yang digunakan pada algoritma semut adalah sebagai berikut: α (alfa) = 5; β (Beta) = 2; ρ (Rho) = 0.3; m (jumlah semut) = 5; Iterasi = 10; T_{ij} = 0.01. Pengujian pencarian rute dengan masukan 6 toko yang dipilih dari 34 toko menghasilkan peta rute distribusi barang dengan rute terpendek, dari gudang melewati 6 toko secara berurutan, dengan hasil rute yang ditunjukkan dalam peta pada gambar 3.



Gambar 3. Hasil dari algoritma semut untuk penyelesaian VRP: informasi peta jalur distribusi barang.

3.2. Percobaan 2.

Percobaan kedua menguji akurasi pencarian rute terpendek algoritma semut terhadap nilai global optimum yang dihasilkan dari algoritma brute force. Pengukuran akurasi dilakukan dengan cara membandingkan keluaran algoritma semut dengan algoritma brute force. Hasil jarak terpendek rute distribusi barang yang dihasilkan dari algoritma semut dibandingkan dengan algoritma brute force untuk 15 kali percobaan dengan masing-masing input toko = 5,6,7,8,9,10. Hasil pengujian tampak dalam tabel 2. Parameter algoritma semut yang digunakan mengacu pada percobaan pertama.

Tabel 2. Jumlah perbedaan hasil algoritma semut dan brute force

	Input 5	Input 6	Input 7	Input 8	Input 9	Input 10
Jumlah perbedaan hasil	0	0	1	2	2	3
Kesamaan hasil = (15- rerata perbedaan hasil)	15	15	14	13	13	12

Tabel 2 menunjukkan jumlah perbedaan dan persamaan hasil jarak terpendek rute distribusi barang antara algoritma semut dengan brute force. Data tabel 2 menunjukkan, untuk input toko = 5 dan 6, tidak ada perbedaan hasil antara algoritma semut dan brute force. Sedangkan untuk input toko = 7,8,9 dan 10 toko terdapat perbedaan hasil. Semakin besar input toko, jumlah perbedaan hasil cenderung meningkat. Sedangkan tabel 3 menunjukkan selisih hasil algoritma brute force dengan algoritma semut. Sebagai contoh untuk input toko = 7, rata rata selisih hasil adalah 4 %, dimana algoritma brute force menghasilkan rute distribusi sejauh = 21500 meter, sedangkan algoritma semut menghasilkan rata rata distribusi sejauh = 22460.

Tabel 3. Rerata selisih hasil hasil algoritma semut dan brute force

	Input 5	Input 6	Input 7	Input 8	Input 9	Input 10
Rerata selisih	0%	0%	4%	4%	5%	4%

Ketika jumlah input bertambah, akan memungkinkan untuk tidak ditemukannya nilai global optimum. Hal ini menunjukkan bahwa hasil dari algoritma semut merupakan sebuah pendekatan. Sesuai dengan teori, algoritma brute force menghasilkan global optimum, sedangkan algoritma metaheuristik menghasilkan sebuah pendekatan hasil yang mendekati global optimum. Algoritma brute force akan *men-generate* seluruh kemungkinan himpunan solusi. Ketika semua himpunan solusi di-*generate* akan didapat solusi yang paling optimum. Namun pembangkitkan semua solusi yang mungkin pada sebuah graph mempunyai kompleksitas NP-hard. Hal ini akan berpengaruh

dengan beban komputasi seiring dengan bertambahnya simpul. Di sisi lain, algoritma heuristik hanya akan membangkitkan himpunan solusi berdasarkan sebuah fungsi metaheuristik dan tidak perlu membangkitkan semua kemungkinan kombinasi solusi. Sehingga hasil algoritma heuristik belum tentu menghasilkan nilai global optimum, namun akan mempunyai beban komputasi yang lebih rendah dari brute force.

3.3. Percobaan 3.

Pada percobaan 2, tabel 2 dan tabel 3 memperlihatkan bahwa akurasi rute terpendek algoritma semut akan berkurang ketika jumlah input toko bertambah. Percobaan ke-3 menguji pengaruh penambahan jumlah semut terhadap efektivitas algoritma semut. Pengujian dilakukan seperti pada percobaan 2 dengan penambahan jumlah semut sebanyak $m=5, 6, 7$ dan 8 . Untuk jumlah semut (m) tersebut, pengujian menghitung jumlah perbedaan hasil antara algoritma semut dan brute force untuk 15 percobaan pada masing-masing input jumlah toko.

Tabel 4. Efektivitas (akurasi) algoritma semut terhadap perubahan parameter jumlah semut

Jumlah perbedaan hasil algoritma semut dengan brute force untuk 15 percobaan tiap input	Input 5	Input 6	Input 7	Input 8	Input 9	Input 10
Untuk jumlah semut $m = 5$	0	0	1	2	2	3
Untuk jumlah semut $m = 6$	0	0	0	0	1	2
Untuk jumlah semut $m = 7$	0	0	0	0	0	0
Untuk jumlah semut $m = 8$	0	0	0	0	0	0

Tabel 4 menunjukkan hasil akurasi rute terpendek algoritma semut terhadap bertambahnya jumlah semut. Untuk jumlah parameter semut = 5 dan 6, bila input jumlah toko ditambah maka akurasi akan menurun. Hal ini terlihat dengan adanya peningkatan jumlah rata-rata perbedaan hasil dengan algoritma brute force. Untuk jumlah parameter semut = 7, percobaan algoritma brute force dan algoritma semut menghasilkan rute dan jarak yang sama. Hal ini menunjukkan bahwa hasil dan jarak rute terpendek dari algoritma sudah akurat, sesuai dengan nilai global optimumnya. Bila jumlah semut ditambah menjadi $m = 8$, hasil tidak akan berubah. Untuk pengujian selanjutnya penambahan jumlah semut sudah tidak memberi pengaruh terhadap akurasi hasil. Berdasarkan percobaan yang dilakukan, efektivitas/kualitas hasil algoritma semut merupakan fungsi dari jumlah semut yang dipergunakan dalam algoritma tersebut. Akurasi akan meningkat sampai batas jumlah semut tertentu. Setelah melewati batas jumlah tertentu sudah tidak akan mempengaruhi hasil akurasi, bahkan bila jumlahnya terlalu banyak akan menambah beban komputasi. Menurut peneliti, tantangan dalam penelitian berikut adalah bagaimana menentukan jumlah semut yang optimum (tepat) dalam sebuah kasus. Dalam penelitian ini, melalui sebuah ujicoba, tampak bahwa untuk input toko sebanyak maksimal 10 buah dengan jumlah parameter semut sebanyak 7 sudah menghasilkan nilai rute global optimum.

Dalam mengambil keputusan pemilihan jalur, seekor semut akan dipengaruhi oleh jumlah intensitas jejak yang dibuat oleh semut sebelumnya ketika melewati rute tersebut. Probabilistik distribusi intensitas jejak sesuai dengan persamaan 1. Semakin banyak intensitas jejak yang ada pada suatu lintasan, maka semut akan cenderung untuk memilih lintasan tersebut. Dengan demikian ketika sampai pada persimpangan, maka semut akan memilih jalur yang jumlah intensitas jejak lebih banyak. Seiring dengan bertambahnya waktu atau iterasi, rute yang sering dilewati oleh banyak semut akhirnya menjadi konvergen dan menjadi sebuah jalur tunggal.

4. KESIMPULAN

Pengujian akurasi algoritma semut dalam penelitian ini menggunakan parameter awal α (alfa) = 5; β (Beta) = 2; ρ (Rho) = 0.3; m (jumlah semut) = 5; Iterasi = 10; T_{ij} = 0.01. Kemudian, jumlah input toko yang diujikan bervariasi berturut-turut adalah = 5,6,7,8,9,10 dan parameter

jumlah semut yang diujikan bervariasi berturut-turut adalah = 5,6,7,8. Pengujian akurasi algoritma semut menggunakan pembandingan algoritma brute force. Pengujian menghasilkan kesimpulan sebagai berikut: akurasi algoritma semut dalam pencarian jarak terpendek pada distribusi barang akan menurun bila jumlah input toko yang dikunjungi semakin bertambah. Akurasi algoritma semut akan meningkat dengan menambah parameter jumlah semut. Dengan jumlah semut yang tepat maka nilai global optimum dapat dicapai. Menentukan jumlah semut yang tepat tidak mudah karena bila jumlah dibawah nilai tersebut akan membuat hasil tidak akurat, namun bila jumlahnya terlalu banyak tentu saja akan menambah beban komputasi.

DAFTAR PUSTAKA

- Christos H. P and Steiglitz.K.,(1982), *Combinatorial Optimization: Algorithms and Complexity*. New Jersey: Prentice-hall.
- Cordeau,J.F., Laporte,G.,Savelsbergh,M.W.,(2007), *Vehicle Routing Handbook* OR&MS.14:pp 367-370.
- Dorigo, M., Maniezzo, V., dan Colomi, A.,(1996), *The Ant System: Optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, andCybernetics—Part B, 26(1), pp.1-13.
- Glover,F dan Kochenberger,G.A.,(2003), *Handbook of Metaheuristic*,Dordrecht, Kluwer Academic Publisher.
- Hung.K. S., Su.F., and Lee.S.J., (2007), *Improving ant colony optimization for solving traveling salesman problem*, Journal of Advanced Computational Intelligence and Intelligent Informatics.
- Lawler.E.L, Lenstra.J.K,Rinnoy-Kan.H.G.,(1985), Eds. *The Traveling Salesman Problem*. New York: Wiley.
- Pop, P. C., Sitar, C. P., Zelina, I., Lupse, V., dan Chira, C., (2011), *Heuristic Algorithms for Solving the Generalized Vehicle Routing Problem*, International Journal Computers, Communication, &Control, 11 (1), hal. 158-165.
- Wei,X.,(2010), *Improved ant colony algorithm based on information entropy*, International Conference on Computational and Information Sciences, IEEE, 987-0-7695-4270-6.